Visualising high-dimensional Pareto relationships in two-dimensional scatterplots

Jonathan Fieldsend and Richard Everson

University of Exeter, Exeter, UK,
{J.E.Fieldsend, R.M.Everson}@exeter.ex.ac.uk

Abstract. In this paper two novel methods for projecting high dimensional data into two dimensions for visualisation are introduced, which aim to limit the loss of *dominance* and *Pareto shell* relationships between solutions to multi-objective optimisation problems. It has already been shown that, in general, it is impossible to completely preserve the dominance relationship when mapping from a higher to a lower dimension – however, approaches that attempt this projection with minimal loss of dominance information are useful for a number of reasons. (1) They may represent the data to the user of a multi-objective optimisation problem in an intuitive fashion, (2) they may help provide insights into the relationships between solutions which are not immediately apparent through other visualisation methods, and (3) they may offer a useful visual medium for *interactive* optimisation. We are concerned here with examining (1) and (2), and developing relatively rapid methods to achieve visualisations, rather than generating an entirely new search/optimisation problem which has to be solved to achieve the visualisation– which may prove infeasible in an interactive environment for real time use. Results are presented on randomly generated data, and the search population of an optimiser as it progresses. Structural insights into the evolution of a set-based optimiser that can be derived from this visualisation are also discussed.

Keywords: Dimension reduction, Pareto optimality, data visualisation.

1 Introduction

The visualisation of a set of solutions maintained by modern evolutionary multi-objective optimisation (EMO) algorithms is of interest to researchers wishing to track the behaviour of algorithms, decision makers who use the output of EMO algorithms, and those wishing to develop *interactive* multi-objective optimisers. Most EMO practitioners are comfortable with visualising a set of solutions with 2 or 3 objective dimensions as a scatter plot of points, and can rapidly determine the non-dominated subset (and those associated with dominated shells [5]) from this. Visualisation of sets with more objectives is often more difficult to interpret via a single scatter plot, and a range of other approaches has been used to visualise these populations in the multi-objective optimisation literature (e.g.

parallel coordinate plots [7, 15, 12] heatmaps [22, 25], directed graphs [24], Chernoff faces [1], and self-organising maps [21, 11]). Dominance relations and shells are not always apparent in these visualisations however (or are only presented between adjacent shells). We are concerned with visualising more than just the estimate of the Pareto front that comes out of most modern EMO algorithms, but more broadly any general set of points (e.g. a search population), from which a visualisation thereof can inform us of the *structure* of the set. Such visualisations can give us extra information relating to the Pareto front estimation, and convey to the problem holder visually *how* an optimisation is progressing.

Here we are concerned with producing a visualisation in the plane, which may be relatively rapidly computed, and is interpretable quickly by both experienced practitioners in EMO, and by problem owners who may not be as familiar with the interpretation of the methods mentioned above. We focus on a single scatter plot of points representing solutions (unlike pairwise coordinate plots [4], which uses $D(D-1)$ separate scatter plots). We shall shortly provide a brief discussion of some existing examples of these, and introduce our two new approaches, but before this we will more formally define Pareto dominance, which is crucial to most modern EMO algorithms, and our visualisation approaches.

2 Pareto dominance

Pareto dominance is used extensively within the search processes of most modern multi-objective optimisation algorithms [4], and, even if not used explicitly in the search process (if aggregation techniques are used for fitness assignment for instance), it is still used to define the properties of the final output set from the optimisers. EMO algorithms are concerned with exploring a *decision space* for design solutions, where an evaluation of a particular design results in an associated point in *objective space*. If we consider (without loss of generality) that all objectives are to be minimised, an objective vector $\mathbf{y}$ of D objectives $(y_1, \ldots, y_D)$ is said to dominate another $\mathbf{y}'$, written $\mathbf{y} \prec \mathbf{y}'$, iff:

$$(y_i \leq y'_i, \forall i) \wedge (\exists i, y_i < y'_i). \tag{1}$$

Succinctly, the best set of solutions to a multi-objective problem (the Pareto set) are the maximal set for which it is impossible (given the problem constraints) to improve any single objective (or group of objectives) of a set member by varying its parameters without having to decrease its performance on one or more other objectives. The image of this set in objective space is known as the Pareto front, $\mathcal{F}$. Given any objective vector set $Y = \{\mathbf{y}_i\}_{i=1}^{N}$, the non-dominated subset of Y is determined as $\mathcal{S}_0 = \{\mathbf{y} \in Y | \nexists \mathbf{z} \in Y, \mathbf{z} \prec \mathbf{y}\}$. This can be taken one step further (as for instance in the popular NSGA-II algorithm [5]), where not only is a dominance relationship put on members of Y (i.e. where any two members are mutually non-dominating, $(\mathbf{y}' \not\prec \mathbf{y}) \wedge (\mathbf{y} \not\prec \mathbf{y}')$, or one dominates the other), but also every member of Y is assigned to a *Pareto shell*. Here members of $\mathcal{S}_0$ are said to be in the zeroth Pareto shell (an estimate of the Pareto front, $\hat{\mathcal{F}}$). Subsequent

shells are defined iteratively in the same manner, subject to the previous shell being removed from Y until the empty set $\emptyset$ is obtained. That is

$$\mathcal{S}_j = \{\mathbf{y} \in Y'_j | \nexists \mathbf{z} \in Y'_j, \mathbf{z} \prec \mathbf{y}\} \tag{2}$$

where $Y'_j = Y \setminus \bigcup_{k=0}^{j-1} \mathcal{S}_k$, and $Y'_j = \emptyset$ for $j \geq k^*$ with some $k^* \in \{1, 2, \ldots\}$. Note that under (1) and (2) it is possible for two members of Y to be mutually non-dominating, but for one to be in a *better* shell than the other.

3 Approaches for visualising multi-dimensional solution sets via scatter plots

If we wish to project an objective vector $\mathbf{y} \in \mathbb{R}^D$ into $\mathbb{R}^2$ to enable visualisation as a point in a plane we must utilise a dimension reduction technique of some form, and, unless there are redundant or perfectly correlated objectives, some information loss is inevitable.

One of the most popular linear dimension reduction techniques is principal component analysis (PCA, [16]), which identifies the directions of objective space that capture the maximum amount of variance in the solutions. Neuroscale [20, 19] has also been used for multi-objective visualisation [11, 8] – but unlike PCA it provides a non-linear mapping. However, although popular across many application domains, both Neuroscale and PCA are oblivious to whether solutions dominate each other, or are mutually non-dominating in multi-objective populations, or what their Pareto shell is. We recently defined a new distance measure, the *dominance distance*, that captures the similarity of the dominance relations of solutions, and we have used this to project mutually non-dominating sets using multi-dimensional scaling [23, 26] to points on the plane [25]. In the same work we also investigated the use of Radviz [13, 14] for this mapping. However even with these representations is is not geometrically apparent which solutions are in which shell or which dominate others.

In [18] a visualisation is presented which does map the $\mathcal{S}_0$ solutions in a multi-dimensional objective space to a mutually non-dominating shell in $\mathbb{R}^2$, with all other mapped solutions being dominated by members of the planar representation of $\mathcal{S}_0$ (although subsequent shells are not explicitly represented). We will discuss the method described in [18] further in Sect. 5, as it is conceptually the close to the methods we propose.

4 Desired properties when visualising shells in the plane

Given a set $Y^D = \{\mathbf{y}_i\}_{i=1}^N \subset \mathbb{R}^D$, we wish to find a mapping to $Y^2 = \{\mathbf{u}_i\}_{i=1}^N \subset \mathbb{R}^2$ such that if $\mathbf{y}_i \prec \mathbf{y}_j$, then $\mathbf{u}_i \prec \mathbf{u}_j$, and if $\mathbf{y}_i \not\prec \mathbf{y}_j$, then $\mathbf{u}_i \not\prec \mathbf{u}_j$. In general a mapping $\mathbf{u} = \mathbf{g}(\mathbf{y})$ with this property does not exist (the reader is directed toward the proof provided in [18] for further details). Instead here we shall concern ourselves with a mapping with two properties, one of which we *can* guarantee, and the second of which we seek a good approximation to, namely:

1. Ensure that the mapping preserves Pareto shells. That is, if we denote by $\mathcal{S}_j^D$ the jth Pareto shell in an ambient space of D dimensions, then $\mathbf{u} \in \mathcal{S}_j^2$ (where $\mathbf{u} = \mathbf{g}(\mathbf{y})$). The superscript on $\mathcal{S}_j$ denotes the dimensionality of the space which it inhabits.
2. Minimise dominance misinformation. We describe three ways to quantify dominance misinformation in Sects. 5, 6 and 7.

Computational methods for quickly determining shells are well-known (see, e.g. [5]) – and are embedded in many EMO algorithms [4]. Furthermore, ensuring that shell members are maintained via a projection into a lower dimension is actually fairly trivial: a very simple approach would be to distribute each shell as illustrated in Fig. 1. Here there are three shells projected from $\mathbb{R}^D$, $D > 2$,

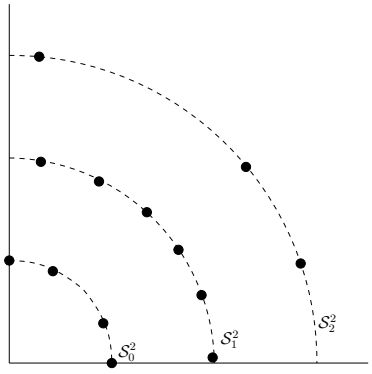

Fig. 1. Simple projection preserving shells but not necessarily dominance relations.

with the number of members in each shell being $|\mathcal{S}_0^D| = 4$, $|\mathcal{S}_1^D| = 6$ and $|\mathcal{S}_2^D| = 3$. When projecting these into $\mathbb{R}^2$ each shell member is projected to a point in the positive quadrant, which lies on the circumference of the circle with radius equal to its shell rank plus one. As long as the mapping is such that the minimum values of the objectives in both dimensions of $\mathcal{S}_j^2$ are greater or equal to the minima in $\mathcal{S}_{j-1}^2$, then this will have the effect that every member of $\mathcal{S}_j^2$ is dominated by at least one member of $\mathcal{S}_{j-1}^2$, and the members of each $\mathcal{S}_j^2$ are mutually non-dominating as required.

This mapping provides the first property mentioned above, but still leads to the issue of *where* to place the $\mathbf{u}_i$ to minimise whichever dominance misinformation objectives may be defined. It is the definition of this property, and methods to incorporate it within a planar visualisation we shall now discuss. The first new approach we consider uses proximity to *domination rays* to convey dominance. The second we introduce uses a direct geometric transference of the dominance relation. First however we will describe the visualisation of Köppen and Yoshida [18].

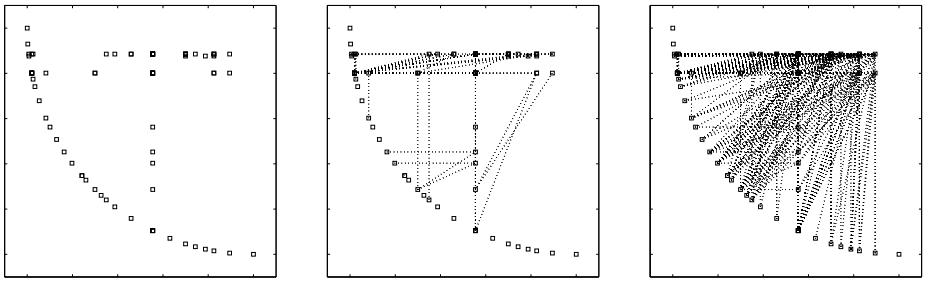

Fig. 2. Visualisation using the approach of [18] of 100 randomly generated points in 4 dimensions. *Left*: Dominance links not shown. *Middle*: Dotted lines show dominance relations between the members of adjacent shells. *Right*: Dotted lines show dominance relations between all members of Y.

5 Visualisation of Köppen and Yoshida

In [18] the non-dominated set from Y^D was mapped to the positive quadrant, lying on the circumference of a circle whose centre point is the origin, and the objectives were maximised. In keeping with the rest of this paper, where objectives are to be minimised, we have 'flipped' the representation from the original and project instead to the negative quadrant. Once the non-dominated subset of Y^D is mapped, for every *dominated* point $\mathbf{y}_i \in Y^D$, the subset of $\mathcal{S}_0^D$ which dominates it is determined, and the worst objective values in the mapping of this set are used to fix the position of $\mathbf{u}_i$ in two dimensions. The exact order of the solutions mapped to $\mathcal{S}_0^2$ was treated as a permutation problem for a multi-objective evolutionary optimiser in [18]. The location of projected solutions on the curve of $\mathcal{S}_0^2$ was determined such that the separation between points was proportional to the distances of immediate $\mathcal{S}_0^2$ neighbours in the original $\mathbb{R}^D$ space. Let $\boldsymbol{\pi}$ be a permutation of the integers $1, \ldots, |\mathcal{S}_0^D|$ describing the order in which the solutions are arranged along $\mathcal{S}_0^2$, so that $\mathbf{u}_{\pi_1}$ is placed on the extreme left, with $\mathbf{u}_{\pi_2}$ next, and so on. Then the two objectives that Köppen and Yoshida seek to minimise in the selection of an optimal permutation are

$$\sum_{k=1}^{|\mathcal{S}_0^D|-1} d(\mathbf{y}_{\pi_k}, \mathbf{y}_{\pi_{k+1}}) \tag{3}$$

where $d(\mathbf{x}, \mathbf{z})$ is the Euclidean distance between $\mathbf{y}_{\pi_k}$ and $\mathbf{y}_{\pi_{k+1}}$, and, denoting by $\mathbf{v}_l \in Y^D$ members of Y^D which are dominated by members of $\mathcal{S}_0^D$,

$$|\{k | \exists 1 \leq i < k < j \leq |\mathcal{S}_0^D|, \mathbf{v}_l \text{ with } (\mathbf{y}_{\pi_i} \prec \mathbf{v}_l) \wedge (\mathbf{y}_{\pi_j} \prec \mathbf{v}_l) \wedge (\mathbf{y}_{\pi_k} \not\prec \mathbf{v}_l)\}| \tag{4}$$

As such, for each element $\mathbf{y}_{\pi_k}$, $1 < k < |\mathcal{S}_0^D|$, (4) checks if *any* two elements lower and higher in the permuted order both dominate a subset of Y^D which the kth ordered solution does not. The minimisation of (3) and (4) is approximated using a real-valued sorting encoding in the NSGA-II algorithm [5]. However, how the

final visualisation permutation is selected from the set of trade-off permutations is not described.

To illustrate this visualisation, we draw 100 points from an isotropic four-dimensional Gaussian distribution, and then map them down to $\mathbb{R}^2$. We first optimise the permutation in the same fashion as [18], using the NSGA-II algorithm, with a population size 100, for 500 generations. We then select the solution on the returned $\hat{\mathcal{F}}$ which minimises (4) as the permutation to use in the final visualisation. We chose this permutation as this objective is essentially a form of dominance misinformation, which is one of the key properties we are concerned with. The resultant visualisation is presented in Fig. 2.

6 Representing dominance in $\mathbb{R}^D$ by *closeness* in $\mathbb{R}^2$

Once we have determined the shell membership of solutions in the original space, the problem is where to place these solutions on their projection to equivalent shells in the lower dimensional space. The first set of novel transformations we present are based upon converting the dominance relation in a higher dimension to a distance relationship in the two dimensional mapping. That is, we attempt to place dominated solutions *close* to those solutions which dominate them, whilst maintaining correct shells. Here we represent the distance to dominating individuals in a different fashion to [18], which does not require the running of a multi-objective optimiser to generate the mapping. Each shell is mapped to a distinct shell (as illustrated in Fig. 1). We then place the solutions, as close as possible to the solutions which dominate them. One way of conceiving of this is that each solution is placed on the curve corresponding to their shell and connected via a spring to all those points which dominate it. These springs act to pull together points which are dominated by the same solutions.

This approach is illustrated in Fig. 3. As in [18] the problem arises as to how to distribute the solutions in $\mathcal{S}_0^2$, however, instead of casting this as a problem to tackle with an evolutionary optimiser, we instead order the solutions using spectral seriation. For a set of $K = |\mathcal{S}_0^D|$ solutions we require a $K \times K$ similarity matrix A describing the similarity between any pair of solutions of this set. Given A, to place similar solutions together, we seek a permutation $\boldsymbol{\pi}$ over the solutions in $\mathbf{S}_0^D$ that minimises:

$$\gamma(\boldsymbol{\pi}) = \sum_{j=1}^{K} \sum_{k=1}^{K} A_{kj}(\pi_k - \pi_j)^2. \tag{5}$$

$\gamma(\boldsymbol{\pi})$ is minimised when similar solutions are placed close to each other, and dissimilar solutions far apart. In general, this is NP-hard because the permutation is discrete [2]. Instead, [2] suggests finding an approximation obtained by relaxing the permutation $\boldsymbol{\pi}$ to a continuous variable $\mathbf{w}$ and minimising:

$$h(\mathbf{w}) = \sum_{j=1}^{K} \sum_{k=1}^{K} A_{kj}(w_k - w_j)^2 \tag{6}$$

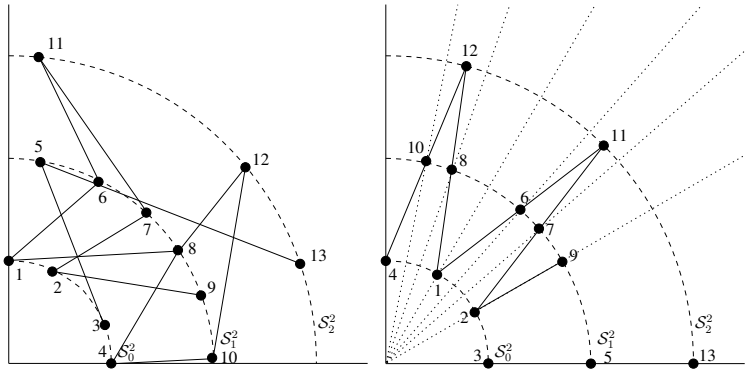

Fig. 3. *Left*: Illustration of the initialisation of a distance-based visualisation, dominance relationships between points on adjacent shells are shown via solid connecting lines. *Right*: Minimal distance rays are plotted projected from the origin through members of $Y \setminus \mathcal{S}_2^2$, which indicate where on each shell a solution must be placed to be the minimal distance away from the dominating point.

with respect to $\mathbf{w}$. This relaxed objective is subject to two constraints. Firstly, to ensure that adding a constant to all w_n does not change the order of the individuals the constraint $\sum_n w_n = 0$ is imposed. Also, in order to avoid the trivial solution in which all $w_n = 0$, we require $\sum_n w_n^2 = 1$. The solution to the constrained problem can be found with linear algebra via the graph Laplacian [10, 3] (further details on how to do this efficiently can be found in [17]). The similarity measure we choose to use here is the *dominance similarity*, which we have used previously for MDS visualisations of multi-objective sets [25, 9].

The dominance similarity between two solutions $\mathbf{y}_j$ and $\mathbf{y}_k$, relative to a third solution $\mathbf{y}_p$, is defined as being proportional to the number of objectives on which $\mathbf{y}_j$ and $\mathbf{y}_k$ have the same relation (greater than, less than, or equal) to $\mathbf{y}_p$. That is:

$$\begin{aligned} S(\mathbf{y}_k, \mathbf{y}_j; \mathbf{y}_p) = \frac{1}{D} \sum_{d=1}^{D} \Big[& I((y_{pd} < y_{kd}) \wedge (y_{pd} < y_{jd})) \\ & + I((y_{pd} = y_{kd}) \wedge (y_{pd} = y_{jd})) \\ & + I((y_{pd} > y_{kd}) \wedge (y_{pd} > y_{jd})) \Big] \end{aligned} \tag{7}$$

where $I(q)$ is the indicator function that returns a value of 1 when the proposition q is true and 0 otherwise.

The dominance similarity across the set $Y = \{\mathbf{y}_i\}_{i=1}^N$ is obtained by averaging $S(\mathbf{y}_k, \mathbf{y}_j; \mathbf{y}_p)$ across all the elements of the set:

$$A_{kj} = \frac{1}{N-2} \sum_{\substack{p=1 \\ p \notin \{k,j\}}}^{N} S(\mathbf{y}_k, \mathbf{y}_j; \mathbf{y}_p). \tag{8}$$

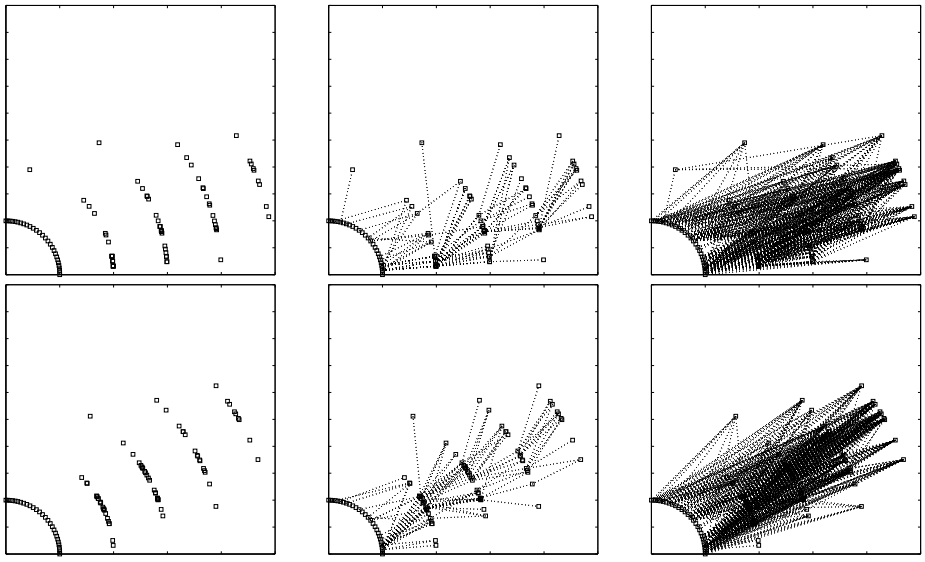

Fig. 4. Visualisation using closeness approach of 100 points randomly generated in 4 dimensions. *Top*: Initial pass. *Bottom*: After refinement iterations. *Left*: No domination links shown. *Middle*: Dotted lines show dominance relations between members of adjacent shells. *Right*: Dotted lines show dominance relations between all members of Y^D.

Utilising (8) to calculate A for just the $\mathcal{S}_0^D$ members of Y^D (but averaging across their similarity to all members of Y^D), gives us an order on the elements of $\mathcal{S}_0^D$ with minimisation of (6), which we transfer to $\mathcal{S}_0^2$. We space the $\mathcal{S}_0^2$ solutions on the curve proportional to their Euclidean distance in $\mathcal{S}_0^D$ (as in [18]).

The distance between shells in the mapping is arbitrary, so we use the angle of the ray passing through a mapped point and the origin to determine the placement of dominated solutions. Specifically, the location of a $\mathbf{u}_i$ is initially placed on the ray through the origin whose angle is the average of the angles of the rays associated with the mapped points which dominate it. As the position of $\mathcal{S}_0^2$ is determined using spectral seriation (as detailed above), the rays defining $\mathcal{S}_1^2$, can be rapidly computed, which, along with $\mathcal{S}_0^2$ can then be used to fix $\mathcal{S}_2^2$, and so on. A schematic of this is in shown the right-hand panel of Fig. 3, and an empirical example is provided in the top panels of Fig. 4 (using the same data as Fig. 2). However, as only the *dominating* points are considered for determining the angle of the ray on which a solution resides, if two solutions are dominated by exactly the same subset of Y^D, then they will lie at the same point – even if the subsets that they both *dominate* are not the same.

In order to resolve the issue of mapping points to the same location when their dominance relationships with Y^D as a whole are not identical, an iterative procedure is used to adjust the locations of $\mathcal{S}_i^2$ points (where $i > 1$), such that the mean of the angles in $\mathbb{R}^2$ of those points which are *dominated* in $\mathbb{R}^D$, as

well as those which dominate in $\mathbb{R}^D$, are used to set the location angles of $\mathcal{S}_i^2$ members. Each shell is evaluated in turn until all the shells have been processed ($\mathcal{S}_0^2$ remaining unchanged). This is repeated until the positions no longer vary. Empirically the number of complete passes before stabilisation is reached has proved small – in the example shown here for instance the location changes were negligible (10^{-3}) within six passes. The bottom panels of Fig. 4 shows the result of this iterative location smoothing – note how a number of individuals in $\mathcal{S}_1^2$ which dominate many elements of $\mathcal{S}_2^2$ have been pulled to a more central region of the $\mathcal{S}_1^2$ shell by this process. On the other hand, the refinement process has left the shells in the same general region as the single pass algorithm, so the single pass seems to give a reasonable approximation (on this instance) to the final refined visualisation.

7 Representing dominance in $\mathbb{R}^D$ by *dominance* in $\mathbb{R}^2$

The second new approach we consider here attempts to *directly* translate the dominance relationships in the higher dimensional space into the two dimensions in a way that is conceptually more akin to [18]. Again, the ordering of solutions mapped to $\mathcal{S}_0^2$ is determined via spectral seriation using dominance similarity, but instead of placing individuals on dominated shells using angles to dominating and dominated solutions, we attempt to minimise the divergence between the dominance relations implied by the lower dimensional visualisation and the true dominance relations in the original space. That is, if an individual $\mathbf{u} = \mathbf{g}(\mathbf{y})$ has the relationship $\mathbf{y}' \prec \mathbf{y}$, then as far as possible we would like $\mathbf{u}' \prec \mathbf{u}$ to hold (and vice versa). To this end we propose a deterministic iterative procedure which attempts to arrange the solutions in each $\mathcal{S}_j^2$ to accomplish this.

When deciding on the placement of the $\mathcal{S}_1^2$ individuals, the members of $\mathcal{S}_0^2$ effectively delimit a number of regions on the feasible curve for $\mathcal{S}_1^2$. Any point in one of these regions has an equivalent dominance relation with $\mathcal{S}_0^2$; that is, any point in a particular curve segment r_k is dominated by the same subset of $\mathcal{S}_0^2$. This is illustrated in the left panel of Fig. 5 – the members of $\mathcal{S}_0^2$ partition $\mathcal{S}_1^2$ into $2|\mathcal{S}_0^2| - 1$ segments into which members of $\mathcal{S}_1^2$ can be placed. In selecting which region to map a solution $\mathbf{y} \in \mathcal{S}_1^D$ to, a natural approach would be to find the one which yields the smallest dominance error. If we denote by $\mathbf{r}_i$ any point in the ith region, and by R_1 the set of these points (one point for each region) for the oneth shell, then we can define a dominance error as having two parts:

$$e_1(\mathbf{r}_i, \mathbf{y}, \mathcal{S}_0^D) = |\{\mathbf{y}' \in \mathcal{S}_0^D | \mathbf{y}' \prec \mathbf{y} \wedge \mathbf{g}(\mathbf{y}') \not\prec \mathbf{r}_i\}|, \tag{9}$$

the number of members of $\mathcal{S}_0^D$ which dominate $\mathbf{y}$ but fail to dominate $\mathbf{r}_i$ in their $\mathcal{S}_0^2$ projection and

$$e_2(\mathbf{r}_i, \mathbf{y}, \mathcal{S}_0^D) = |\{\mathbf{y}' \in \mathcal{S}_0^D | \mathbf{y}' \not\prec \mathbf{y} \wedge \mathbf{g}(\mathbf{y}') \prec \mathbf{r}_i\}|, \tag{10}$$

the number of members of $\mathbf{S}_0^D$ which do not dominate $\mathbf{y}$ but incorrectly dominate $\mathbf{r}_i$ in their $\mathbf{S}_0^2$ projection. Empirically we find that simply summing these two

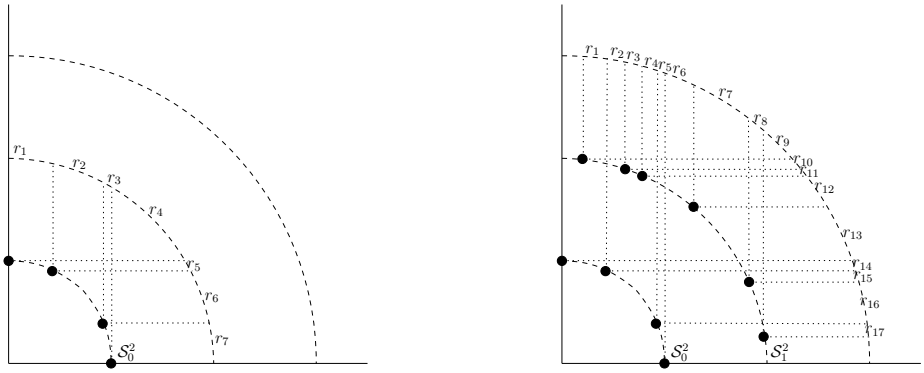

Fig. 5. *Left*: Illustration of potential intervals for placement of $\mathcal{S}_1^2$ individuals, once the order of $\mathcal{S}_0^2$ has been determined. *Right*: Illustration of potential intervals for placement of $\mathcal{S}_2^2$ individuals, once the order of $\mathcal{S}_1^2$ has been determined. Note there are no intervals on the extremes, as the interval ranges must be dominated by at least one member of the previous shell.

penalty terms to generate a combined error (to minimise) does not lead to a satisfying projection. This is because (10) tends to outweigh (9) with the result that all the solutions in dominated shells tend to be pushed close to the axes. Therefore, we find the subset of R_j which minimises e_1, and then choose the element of this subset with the lowest e_2. By geometry, we can see that $e_1 = 0$ can be achieved for *any* dominated element Y^D projected onto $\mathcal{S}_i^2$, as long as the shell radius for $\mathcal{S}_i^2$ is $\sqrt{2}$ times the shell radius for $\mathcal{S}_{i-1}^2$, or greater. This will mean that there is a region on the $\mathcal{S}_i^2$ curve which is dominated by *all* elements of $\mathcal{S}_{i-1}^2$, therefore we choose the shell radii accordingly to guarantee this.

It is also possible (and indeed inevitable if $2|\mathcal{S}_0^D| - 1 < |\mathcal{S}_1^D|$)[1] for some solutions in $\mathcal{S}_1^2$ to be placed in the *same* region. We would not however wish to place them on exactly the same point, as they may not dominate the same subset of Y^D. If more than one solution is placed in a region, then they are spaced evenly across the curve segment that region defines, otherwise it is placed in the centre of the segment. After the $\mathcal{S}_1^2$ shell is assigned, subsequent shells are assigned in order in a similar way to that described for $\mathcal{S}_1^2$; that is in (9) and (10) $\mathcal{S}_0^D$ is replaced by $\bigcup_{k=0}^{j-1} \mathcal{S}_k^D$ (where the jth shell is being assigned).

This still leaves the problem of how to order multiple solutions mapped to the same region. This is, however, another permutation problem, and as such we simply construct the dominance similarity matrix for solutions in this region, and order them according to the order suggested by spectral seriation.

[1] Generally, in the jth shell ($j > 0$), there are *at most* $1 + \sum_{i=0}^{j-1} 2(|\mathcal{S}_i^2| - 1)$ regions where a shell member may be placed, this growth is illustrated in the right panel of Fig. 5.

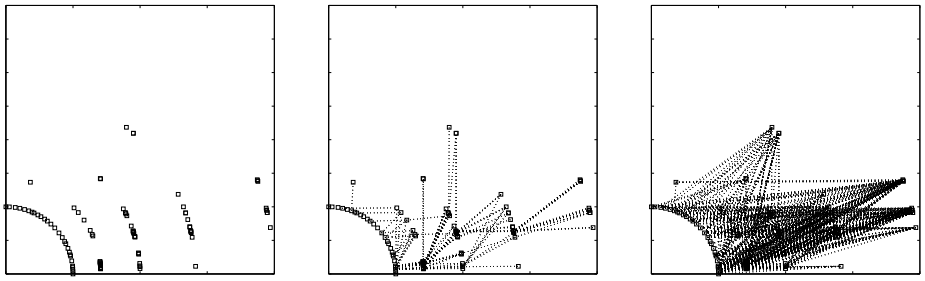

Fig. 6. 100 randomly generated points in 4 dimensions as in previous illustration, visualisation using dominance approach but with modified minimisation function. *Left*: Dominance links not shown. *Middle*: Dotted lines show dominance relations between members of adjacent shells. *Right*: Dotted lines show dominance relations between all members of Y^D.

Table 1. Property comparison of the three scatter plot visualisation methods. $\mathbf{y}_i$ and $\mathbf{y}_j$ are original objective vectors drawn from Y^D, and $\mathbf{u}_i$ and $\mathbf{u}_j$ are their corresponding projections using the methods examined into $\mathbb{R}^2$.

	Köppen & Yoshida	Distance-based	Dominance-based
(I) If $\mathbf{y}_i \in \mathcal{S}_k^D$ then $\mathbf{u}_i \in \mathcal{S}_k^2$	✗	✓	✓
(II) If $\mathbf{y}_i \prec \mathbf{y}_j$ then $\mathbf{u}_i \prec \mathbf{u}_j$	✗†	✗	✓
(III) If $\mathbf{y}_i \not\prec \mathbf{y}_j$ then $\mathbf{u}_i \not\prec \mathbf{u}_j$	✗	✗	✗

† If solutions in $\mathcal{S}_0^2$ can be arranged so that (4) is equal to zero, then (II) holds for any pair of points which are not mapped to the same location in $\mathbb{R}^2$. If (4) is not equal to zero then (II) cannot be guaranteed to hold anywhere in the mapping of [18].

The visualisation approach, using our running example, leads to the projection shown in Fig. 6. All linked points in this visualisation can be seen to dominate in a geometric sense.

8 Visualisation comparisons

The dominance and shell properties of the three visualisations we have illustrated here (that of [18], and our two new visualisations) are presented in Table 1. Assuming a permutation of $\mathcal{S}_0^2$ can be found such that (4) is equal to zero, then the method of [18] guarantees property (II) through the placement of the dominated solutions in Y^2 using the worst values of the mapped dominating subset of Y_0^D. In practice an ordering which obtains (4) equal to zero is rare however, and it still allows points to be placed on the same location when one dominates the other. Our distance/angle-based visualisation guarantees (I), however as it reinterprets geometric dominance into angles it does not attempt to provide (II) or (III). Our dominance-based visualisation guarantees both (I) and (II), and tries to minimise (III) (subject to (II)), by minimising (10) and the corresponding

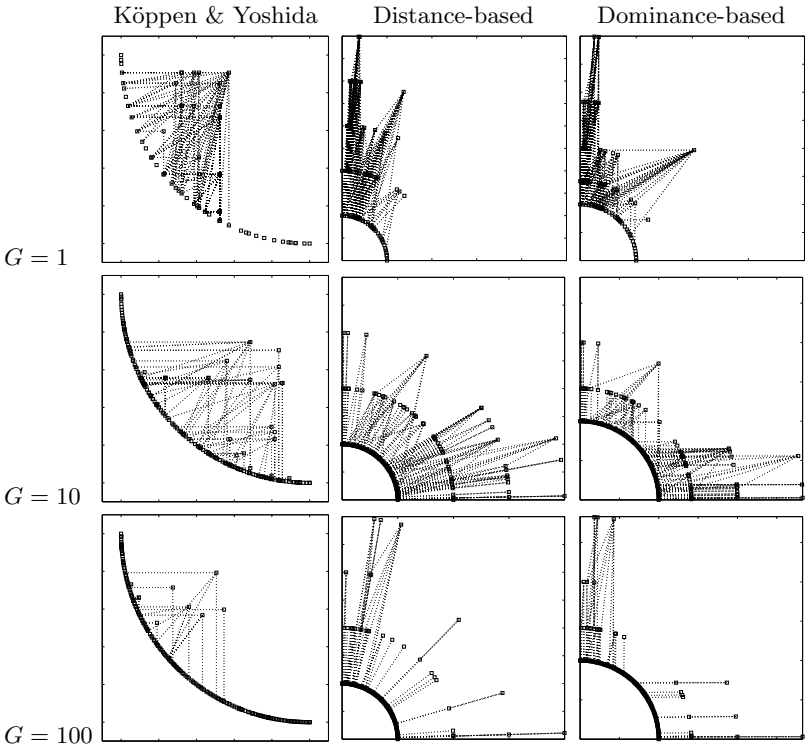

Fig. 7. Visualisation of SPEA2 4-objective problem search populations. G indicates the generation. Dotted lines show dominance relations between *all* members of Y^D.

objective functions for later shells. Note, other mappings to the plane previously used in the EMO field (e.g. PCA, MDS, Neuroscale, RadViz) do not guarantee *any* of the properties listed in the Table 1.

We now provide a further brief comparison of the two methods we have introduced here, along with the method of [18], using the run time population of an EMO algorithm. We visualise the combined archive and search population of the popular SPEA2 algorithm [27] as it progresses through the optimisation of a 4-objective optimisation problem (the DTLZ2 test problem [6]). The algorithm is run with an archive size of 100 and a population size of 100, and we visualise the combined population of 200 solutions after 1, 10 and 100 generations in Fig. 7. A number of structural properties are immediately apparent from the runtime results presented in Fig. 7. The two visualisations introduced here clearly show the number of shells, and the proportion of points on these can be reasonably gauged. It is interesting to note that the method for placing dominated individuals from [18] visually loses many dominated individuals in the population entirely because it maps them onto the same location as a solution which dominates them. On the other hand, [18] does push the elements of $\mathcal{S}_0^2$ (the projection of the estimated Pareto front, $\hat{\mathcal{F}}$) which do not dominate *any* other set members

to the two extremes of the shell, so it is clear which non-dominated members are structurally unsupported. This is not so immediately determined from the other two visualisations, however it can be coarsely judged by looking at the number of $\mathcal{S}_0^2$ members which do not have lines attached to them.

All visualisation approaches show that the number of dominated points in the search population is decreasing as the search progresses, indicating that the search population is spreading out and advancing slowly (rather than making big jumps forward – which would lead to a larger proportion of Y being dominated).

The distance- and dominance-based visualisations could be modified to use more of the plotting space, by making the dominance errors (used in fixing point locations) concerned with only relationships between *adjacent* shells – however this would reduce the structural inference possible from the plots. For example, it can be seen in the dominance-based visualisation that at $G = 1$, the members of shell 4 (and all bar one member of shell 3) are exclusively dominated by only a small number of members in shell 0, as the members of these shells are gathered to the top left of the plot, and property (II) means that only members of shell 0 below and to the left of them can dominate them. This kind of structural information is not readily apparent in the method of [18], and completely lacking from approaches which attempt to visualise $\hat{\mathcal{F}}$ alone.

9 Discussion

We have introduced two related novel visualisations of multi-dimensional sets of points, which endeavour to preserve Pareto shell and dominance information. As with all point mappings which reduce the dimensionality of the data, there is inevitably some loss in information, and assessing the quality of the visualisations presented is by its nature subjective. However, we believe they are a useful contribution to the group of methods in the field; because they exhibit some useful properties (listed in Table 1), and have an advantage over some other approaches in their speed of computation. Of the two, we have a slight preference for the dominance-based approach. It guarantees two useful relationships in Y^D are preserved in Y^2 and endeavours to translate the geometric properties practitioners are already familiar with. That said, as long as the user is comfortable inferring dominance by angle similarity (and/or links), then the angular/distance-based approach is generally quicker to compute (as all the various candidate $\mathbf{r}$ of the dominance method do not need to be computed and compared). The method of Köppen and Yoshida has the advantage of being compact, however shell information is lost and it can become more cluttered than the other two. It is also expensive to generate, as (4) is not quick to compute and the ordering of solutions in $\mathcal{S}_0^2$ requires the use of an evolutionary optimiser. We note however that spectral seriation could be used to obtain a permutation for $\mathcal{S}_0^2$ here also.

It is possible that 'better' point locations may be found given the fitting objectives of our two methods using evolutionary optimisation approaches, however this would likely undermine their speed benefits if used to visualise search populations *during* a multi-objective optimisation. There are however further avenues

of research that may prove useful. There may be useful information that can be conveyed in the magnitude of the shell radii (as used in [18] to convey the range and magnitude of $\mathcal{S}_0^D$). The shape of the shells being mapped to is also arbitrary; by allowing a greater freedom in location we may be able to convey more information, and it may also improve the false positive rate if some solutions can be *closer* to their dominating shell and therefore be erroneously dominated by fewer solutions.

We also look forward to examining the use of these visualisation approaches in *interactive* optimisation, for example using the structural information presented to select population members for further examination and/or variation.

Acknowledgements and resources

The authors would like to thank the anonymous reviewers for their very useful and constructive comments. MATLAB code for the work presented here can be obtained from http://emps.exeter.ac.uk/staff/jefields/.

References

1. G. Agrawal, K. Lewis, K. Chugh, C-H. Huang, S. Parashar, and C. L. Bloebaum. Intuitive Visualization of Pareto Frontier for Multi-Objective Optimization in n-Dimensional Space. In *proceedings of 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.
2. J. E. Atkins, E. G. Boman, and B. Hendrikson. A Spectral Algorithm for Seriation and the Consecutive Ones Problem. *SIAM Journal on Computing*, 28(1):297–310, 1998.
3. F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
4. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001.
5. K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.
6. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Multi–Objective Optimization Test Problems. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 825–830, 2002.
7. M. D'Ocagane. *Coordonnées parallles et axiales: Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèlles*. Gauthier-Villars, reprinted by Kessinger Publishing, 1885.
8. R. M. Everson and J. E. Fieldsend. Multi-class ROC Analysis from a Multi-objective Optimisation Perspective. *Pattern Recognition Letters*, 27:531–556, 2006.
9. R. M. Everson, D. J. Walker, and J. E. Fieldsend. League Tables: Construction and Visualisation from Multiple Key Performance Indicators. Technical report, The University of Exeter, 2012.
10. M. Fiedler. Algebraic Connectivity of Graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

11. J. E. Fieldsend and R. M. Everson. Visualisation of Multi-class ROC Surfaces. In *Proceedings of the ICML 2005 Workshop on ROC Analysis in Machine Learning*, pages 49–56, 2005.
12. C. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. Morgan Kauffman, 1993.
13. P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. DNA Visual and Analytic Data Mining. In *VIS'97: Proceedings of the 9th Conference on Visualization*, pages 437–441, Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.
14. P. E. Hoffman. *Table Visualisation: a Formal Model and its Applications.* PhD thesis, University of Massachuesetts Lowell, 1999.
15. A. Inselberg. N-dimensional Coordinates. In *Picture Data Description & Management, IEEE PAMI*, page 136, 1980.
16. I. T. Jolliffe. *Principal Component Analysis.* Springer, 2002.
17. A. Kaveh and H. A. Rahimi Bondarabady. Finite Element Mesh Decomposition Using Complementary Laplacian Matrix. *Communications in Numerical Methods in Engineering*, 16(379-389), 2000.
18. M. Köppen and K. Yoshida. Visualization of Pareto-sets in evolutionary multiobjective optimization. In *Proceedings of the 7th International Conference on Hybrid Intelligent Systems*, pages 156–161, Washington, DC, USA, 2007. IEEE Computer Society.
19. D. Lowe and M. E. Tipping. Feed-Forward Neural Networks and Topographic Mappings for Exploratory Data Analysis. *Neural Computing and Applications*, 4(2):83–95, 1996.
20. D. Lowe and M. E. Tipping. Neuroscale: Novel Topographic Feature Extraction Using RBF Networks. In *Advances in Neural Information Processing Systems 9*, NIPS'96, pages 543–549, 1996.
21. S. Obayashi. Pareto Solutions of Multipoint Design of Supersonic Wings using Evolutionary Algorithms. In *Adaptive Computing in Design and Manufacture V*, pages 3–15. Springer-Verlag, 2002.
22. A. Pryke, S. Mostaghim, and A. Nazemi. Heatmap Visualization of Population Based Multi Objective Algorithms. In *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization*, pages 361–375. Springer-Verlag, 2007.
23. J. W. Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, 18(5):401–409, 1969.
24. D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualisation and Ordering of Many-objective Populations. In *IEEE Congress on Evolutionary Computation*, pages 3664–3671, July 2010.
25. D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualising Mutually Nondominating Solution Sets in Many-objective Optimisation. *IEEE Transactions on Evolutionary Compuation*, 2012. To appear, available from http://cis.ieee.org/.
26. A. R. Webb. *Statistical Pattern Recognition, 2nd Edition.* John Wiley & Sons, 2002.
27. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report TIK-Report 103, Swiss Federal Institute of Technology Zurich (ETH), May 2001.